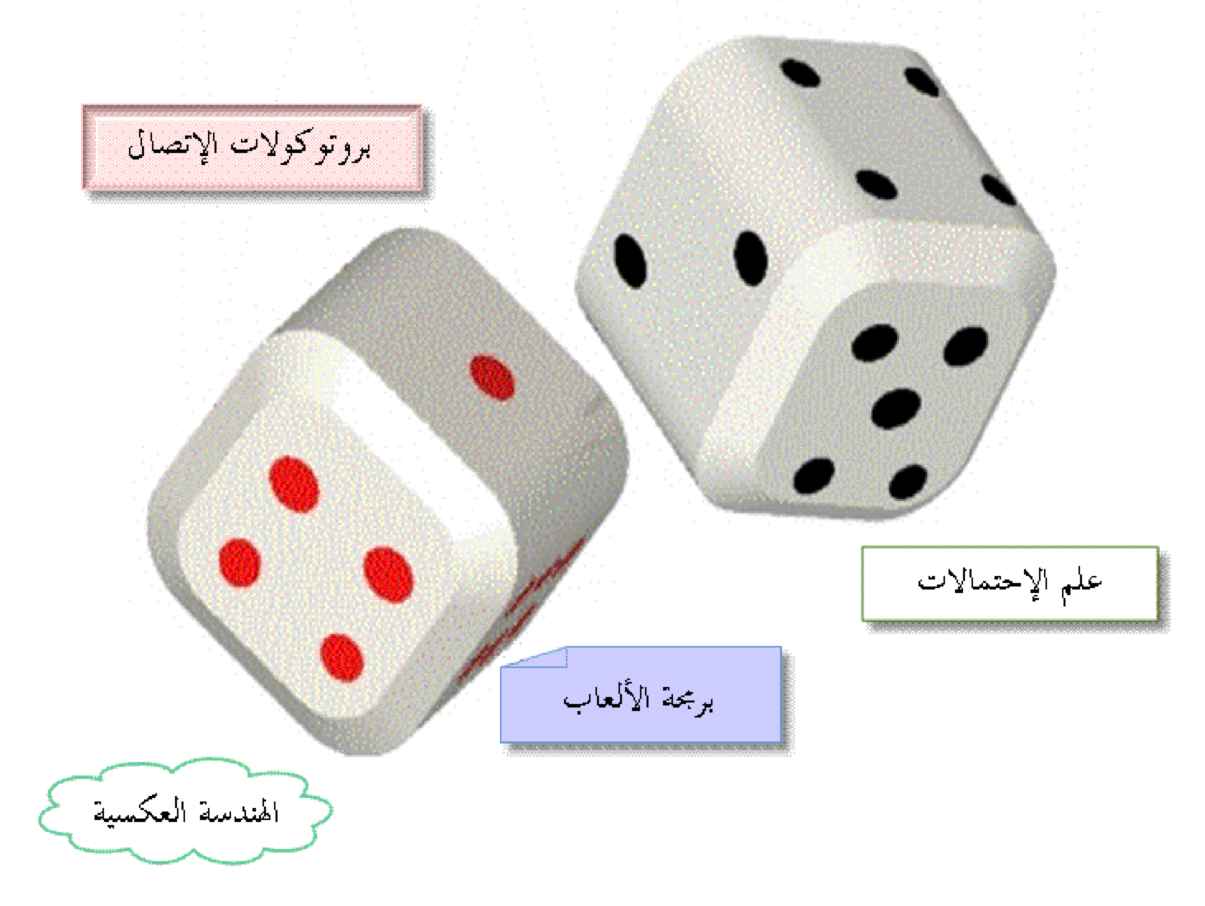
توليد الأرقام العشوائية !!!

(نظرة تحليلية مفصلة)



تأليف: أحمد/محمد (المتألق)
€1M0µT£eELI9

أن تعرف البرمجة فهذا كل شيء ....



الحمد لله الذي بحمده يُستفتح كل كتاب و بذكره يُصدر كل خطاب وبفضله يتنعم أهل النعيم في دار الجزاء و الثواب والصلاة و السلام على سيد المرسلين و إمام المتقين المبعوث رحمة للعالمين محمد ابن عبد الله الصادق الأمين و على صحابته الأخيار و من تبعهم بإحسان إلى يوم الدين أما بعد :

قُمت بتأليف هذا الكتاب كهدية متواضعة مني إلى منتديات العاصفة و منتديات الفريق العربي للبرمجة وقد حاولت أن أجيب فيه عن كل التساؤلات المطروحة بشأن الأعداد العشوائية .. انطلاقا من ماهيتها مرورا بكيفية توليدها و انتهاء بأهم الخوارزميات المستخدمة في هذا المجال .. ليكون –بإذن الله– مرجـــعا لمن أراد التعرف أكثر على الأعداد العشوائية.

إذا رأيت أي خطأ أو تقصير في الكتاب فاعلم أن ذلك من نفسي و من الشيطان فالكمال لله و حده عز و جل.

إذا قرأت كتابي وانتفعت به *** فاحذر وقيت الردى من أن تغيره

ودعه لي سالما إني شغفت به *** لولا مخافة كتم العلم لم تره



بداية .. وكالعادة .. أُهدي هذا الكتاب إلى شقيقي و حبيبي الغالي chikoo و إلى أختي الأكبر zinat و إلى أستاذي المحترف sembawyo الذي تربيت على يديه و تعلمت منه الكثير من أساليب البرمجة فجزاه الله ألف خير .. و أسأل الله سبحانه و تعالى أن يوفقه في الدارين و يزيده من فضله .. إنه ولي ذلك و القادر عليه,كما أُهدي هذا الكتاب إلى المشرف المتميز الحضراني و هو مشرف منتدى حماية البريد و الأجهزة في منتديات العاصفة و المراقب العام و إلى المصمم المحترف "المخترق" وهو من كبار أعضاء منتديات العاصفة و مشرف منتدى التصميم و الجرافيكس و إلى أُستاذي الفاضل الشمري الذي يُشرف حاليا على منتدى الألعاب والجرافيكس في منتديات الفريق العربي للبرمجة.

عن المؤلف:

الإسم:أحمد/محمد

اللقب: المتألق (elmoute2eli9)

سنة الميلاد:1992

الدولة:موريتانيا

الهواية: programming & hacking

المستوى:طالب جامعي بكلية العلوم و التقنيات.

للاستفسار و تبادل الخبرات: elmoute2eli9@hotmail.com

صورة المؤلف:



المراجع التي اعتمدت عليها:

1-موضوع "الدوال العشوائية .. هل هي فعلا عشوائية؟" للأستاذ mohamed clay في مجتمع لينوكس العربي.

2-موضوع "توليد أرقام عشوائية" للأستاذ Mouradpr في الفريق العربي للهندسة العكسية.

في هذا الكتاب سنتعرف على النقاط التالية:

1- ما هو العدد العشوائي ؟ وهل هو موجود ؟
2- لماذا نحتاج لتوليد الأرقام العشوائية ؟
3- كيف يمكننا توليد أرقام عشوائية (باستخدام الدوال الجاهزة) ؟؟؟
4-كيف ننشئ دالة عشوائية ؟

لنبدأ على بركة الله ...

1-ما هو العدد العشوائي ؟ وهل هو موجود ؟

أولا .. عندما نقول "عدد عشوائي" فهذا لا يعني بالضرورة أنه عشوائي بالمعنى العام .. لا لا لا .. العدد العشوائي هو العدد المولد من طرف دالة مجهولة السلوك (نسبيا) و عندما أقول نسبيا فأنا أقصد "المستخدم" , أي أن المستخدم عندما يقوم بتشغيل الدالة ستقوم بإرجاع عدد عشوائي كما يتصور "هو" و لكن في الحقيقة .. فإن هذا العدد العشوائي يخضع لقانون معين ..المبرمج فقط هو من يعرفه !

ألا ترى بعض التناقض في كلامنا السابق ! قلنا أن العدد "عشوائي" ثم قلنا بأنه يخضع لقانون معين !!! كيف ؟؟؟

بداية .. أحب أن ألفت انتباهك إلى الجدل الدائر حول العدد العشوائي !!! فالمبرمجون أرادوا برمجة دالة تمكن من توليد أرقام عشوائية ! وكانت المشكلة تكمن في برمجة الدالة العشوائية .. فنحن حين نقول "دالة مبرمجة" فهذا يعني أن الدالة تخضع لقانون محدد .. أما العدد العشوائي فلا يخضع لأي قانون .. وإلا فسيكون عددا محددا و سيفقد عشوائيته .. هنا تكمن المشكلة ...

السؤال المطروح الآن هو:

إذا استطعنا إخضاع كل شيء لقانون محدد فهل هذا يعني بأن العشوائية موجودة أصلا أم ...؟

في الحقيقة .. لا يوجد شيء اسمه (العشوائية) لأن الكون لا يمكنه الاستمرار لأكثر من مليار عام و الأرض لأكثر من 60 مليون عام إذا كان هناك شيء "عشوائي" و لو كان صغيـــــــــــــــرا!!!

هناك تأثير يسمي (تأثير الفراشة) تم تجسيده في فيلم أجنبي بنفس الاسم و هو ببساطة يقول أنه لو تم تغير شيء صغير جدا جدا .. في الماضي و لو كان مجرد " فراشة " فإن هذا سيؤثر قطعاً بشكل ضخم جدا جدا .. في الحاضر و المستقبل.

لو طبقنا هذا التأثير على (كلمة عشوائي) لوجدنا أنه لن يصبح الكون كوناً إذا كان هناك شيء اسمه العشوائية.

أذكر هنا أيضاً (دالة ليمان) التي لم يجد العلماء لها حلا منذ 150 عاماً و هي دالة رياضية حاول العالم (ليمان) أن يربط فيها بين عدد الأرقام الأولية في مجموعة معينة من الأرقام تبدأ بالصفر و بين نهاية أو آخر رقم في هذه المجموعة.

الدارس لهذه الدالة يجد أول الأمر أنها أرقام عشوائية و لكن إصرار العلماء على وضعها كدالة و لها حل و يحاولون بكل جهدهم إيجاد هذا الحل .. يؤكد على عدم اقتناع العلماء أصلاً بوجود العشوائية.

و يرى البعض أن كلمة "عشوائي" هي مرادف لكلمة "مجهول" فنحن عندما نعجز عن تفسير ظاهرة معينة نقول أنها عشوائية .. مثلا:"النرد" فهناك من يعتقد أنها ظاهرة عشوائية بشكل مطلق !!! ولكنني أرى أنه لو توفرتْ لنا بعض البارمترات فيمكننا حينها توقع نتيجة الرمي .. فمثلا .. لو عرفنا كتلة النرد والسرعة التي قُذف بها والوضعية التي كان عليها فإننا قد نتوقع الرقم الذي سنحصل عليه بعد رمية !!!

2-لماذا نحتاج لتوليد الأرقام العشوائية ؟

تُستخدم الأعداد العشوائية في الكثير من الأمور .. نذكر منها على سبيل المثال لا الحصر:

a-علم الإحتمالات.

b-برمجة الألعاب.

c-الهندسة العكسية.

d-بروتوكولات الإتصال,مثل بروتوكول TCP.

a-علم الإحتمالات:

سأعطي مثال بسيــــــــــــط .. لكي تتضح الفكرة .. لدينا مجموعة من الكرات في صندوق .. و نريد سحب إحدى هذه الكرات بشكل عشوائي!

كيف ؟؟

الأمر بسيــــــــــــط للغاية .. كلما ما علينا فعله هو ترقيم هذه الكرات و أخذ عدد عشوائي من بين الأعداد المتاحة .. لاحظ أن كل رقم تقابله كرة مما يعني أننا سحبنا إحدى الكرات بشكل عشوائي.

b-برمجة الألعاب.

في الحقيقة .. فإن الألعاب مجال خصب للبحث عن خوارزمية تولد أرقام عشوائية .. فمثلا .. لكي لا يُصر اللاعب على أن يبرمج شكل واحد للعبة معينة فيمكنه أن يعيد نفس اللعبة بعدة أشكال !!! لنأخذ مثال:

حين نلعب مثلا لعبة "Tick-Tack" فالنجمة تظهر كل مرة في مكان عشوائي .. وهذا مما يجعل اللعبة أكثر إثارة.

نفس الفكرة يمكننا تطبيقها في الكثير من الألعاب الأخرى ...

c-الهندسة العكسية:

أما في مجال الهندسة العكسية فنحن نحتاجها مثلا من أجل برمجة "الكيجن".

للذين لا يعرفون ما هو "الكيجين":

هناك طرق كثيرة لحماية البرامج و من أسهلها استخدام أرقام سرية معينة يتم تحديدها من قبل الشركة المنتجة للبرنامج .. وهذه الأرقام قد تعتمد في تحديدها على اسم المستخدم أو بريده الالكتروني أو الرقم التسلسلي للجهاز أو أي معلومة أخرى أو مجموعة من هذه المعلومات معا!! و عادة تحصل على هذا الرقم عند شراء المنتج من الشركة و تعطيهم بياناتك فيعطونك الرقم السري (السيريال) الذي يتيح لك استخدام البرنامج بشكل كامل.

نأتي حاليا لكيفية كسر هذه الحماية .. وأشهر طريقة هي استخدام "الكيجين".

و لكن ما هو "الكيجين" ؟؟؟

الكيجين (keygen):هو اختصار لـــ key generator أي مولد المفاتيح و هو عبارة عن برنامج صغير يطلب منك نفس المعلومة التي تعتمد عليها الشركة في إعطاء الرقم السري !! و يولد لك الرقم الذي تستطيع فتح البرنامج به .. عادة ما يستطيع بعض المبرمجين معرفة الخوارزميات التي تعتمد عليها هذه الشركات في توليد الأرقام السرية و ينشئون برنامج ليستخدمه الناس!! و طريقته سهلة (نسبيا) فهي مثلا تطلب منك الاسم الموجود في جهازك و من ثم تعطيك الرقم الذي تنسخه في البرنامج فيعمل معك بشكل كامل.

نعود مرة أخرى لموضوعنا السابق و هو فائدة الأعداد العشوائية من أجل برمجة الكيجين .. تخيل أن خوارزمية التحقق من سيريال برنامج ما .. تخضع للشروط التالية:

الشرط الأول:

السيريال يجب أن يتكون من 10 حروف.

الشرط الثاني:

يقارن الحرف الأول مع A والحرف الثامن مع S والحرف العاشر مع D.

إذا لحل هذه الخوارزمية يكفي أن نأخذ سيريال عشوائي من 10 حروف بحيث يحقق الشروط .. ليكن هذا السيريال مثلا: AFGMRTWSOD

هذا السيريال يحقق الخوارزمية ولكن يبقى السؤال هو:

لماذا لا نكتفي بنشر سيريال واحد يحقق الخوارزمية.

السبب هو : مثلا .. أنت تعبت حتى استنتجت الخوارزمية ثم نشرت السيريال و من ثم قام شخص آخر بسرقة هذا السيريال وقال بأنه ملك له !!!

من نصدق ؟؟؟ أكيـــــــــــد سنصدق من فهم الخوارزمية.

لذلك وجب علينا برمجة "الكيجن" الذي سيولد عدة سيريالات تحقق خوارزمية التحقق ...

الكلام السابق يكون صحيحا إذا كان الشخص لا يعرف الخوارزمية أما إذا كان يعرفها فأنتم إذا سواء !!!

بالمناسبة فإن الأعداد العشوائية تستخدم بشكل كبير في خوارزميات التشفير لأن أغلب هذه الخوارزميات تحتاج إلى تزويدها بأرقام عشوائية لتبدأ عملها بحيث لا يمكن التكهن بنتائج الخوارزمية و لا مُدخلاتها الأصلية حتى لمن يعرف جيدا طريقة عملها !!!

ملاحظات:

*)عادة ما تكون الخوارزميات التي تتعلق بالكيجين معقدة جدا جدا .. بحيث يكون من الصعب كسرها .. ولكن هذا لا يمنع البعض من استخدام مهاراته البرمجية (للأسف الشديد) من أجل كسر هذه الخوارزميات.

**)لسنا هنا لنشرح طريقة "كسر خوارزميات الكيجين" , أبدا معاذ الله .. فأنا لست "قاتلا مأجورا" إنما أردت (فقط) توضيح بعض المسائل التي تتعلق بالأعداد العشوائية.

d-بروتوكولات الإتصال,مثل بروتوكول TCP.

الحصول على أرقام عشوائية أمر مهم جدا في علوم الحاسوب خاصة في مجال بروتوكولات الإتصالات .. مثلا في بروتوكول TCP هناك رقم تسلسلي لكل حزمة يسمح لطرف الإتصال الآخر بترتيب الحزم عند وصولها و بمعرفة ما ضاع منها و إعادة إرساله .. و هناك هجوم كلاسيكي يتمكن خلاله طرف ثالث من الدخول في الإتصال و تحويله إذا كان بإمكانه التكهن بالرقم التسلسلي المستعمل باختصار إذا كنت تعرف نوع الخوارزمية و كانت الدالة العشوائية التي تزودها ضعيفة بحيث يمكن التكهن بالأرقام العشوائية التي تنتجها فيمكنك نظريا كسر هذه الخوارزمية.

الحصول على أرقام حقيقة عشوائية صعب جدا في الحواسيب ! لأن هذه الأخيرة صممت كي تكون محددة في عملها (deterministic) و بالتالي أي دالة رياضية تستخدم لتوليد أرقام عشوائية سيتم كسرها و لو طالت دورتها .. لذلك عندما يحتاج برنامج ما إلى أرقام عشوائية تصلح في عمليات التشفير

(crypto grade random numbers) فيجب تزويد الحاسوب ببطاقة خاصة تقوم بإنتاج هذه الأعداد انطلاقا من "ظاهرة فيزيائية عشوائية" و ليس "دوال رياضية" و نظام FreeBSD يدعم هذا النوع من العتاد.

3-كيف يمكننا توليد أرقام عشوائية ؟؟؟

يمكن توليد الأرقام العشوائية بأكثر من (قانون,خوارزمية,طريقة) و كلما كان القانون أكثر تعقيدا يكون العدد أقرب ما يكون للعشوائية .. يعني يصبح من المستحيل (بالنسبة للمستخدم) معرفة سلوك الدالة التي تتولى المهمة .. فنحن عندما نقوم بإنشاء دالة عشوائية ترجع قيمة العدد المُدخل قسمة على 2 !!! الآن أصبح من السهل اكتشاف سلوك هذه الدالة .. فعندما يقوم المستخدم بتشغيل هذه الدالة 10 مرات (مثلا) سيتمكن من استنتاج سلوك هذه الدالة بكل سهولة .. هنا يكمن احتياجنا في قانون رياضي يكون معقد بعض الشيء .. مما يربك المستخدم أثناء محاولته معرفة سلوك الدالة.

الكلام السابق يكون أكثر دقة إذا كان العدد العشوائي عبارة عن عدد "كسري" وكلما كانت الفاصلة غير منتهية كان الكلام أدق .. أما إذا كان العدد العشوائي عبارة عن عدد صحيح .. فسيكون الكلام السابق غير دقيق تماما ! لأن المستخدم أصبح بإمكانه اكتشاف سلوك الدالة التي تولد الأعداد العشوائية !!! فيمكنه (مثلا) توليد 1000 عدد بين 1 و 10 مما يقوده إلى استنتاج سلوك الدالة .. وهكذا .. يعني يختار مجال ضيق ويقوم بتوليد الكثير من الأعداد العشوائية في المجال المُختار .. مُستنتجا الملامح الأولى لسلوك هذه الدالة.

الآن .. سنقوم باستعمال دالة تقوم بتوليد رقم عشوائي.
الدالة التي سنستعملها هي الدالة rand وتوجد في المكتبة cstdlib , هذه الدالة تتبع خوارزمية مُعقدة بعض الشيء !!! مما يجعل عملية تتبع الرقم المُولد من طرف الدالة "عملية صعبة" إن لم تكن "مستحيلة" , هذا في حالة استخدام البذرة أما إذا لم نستخدمها فستقوم الدالة بتوليد أرقام عشوائية "شبه منتظمة" و السبب هو أنها تتبع خوارزمية ثابتة لتوليد الأرقام العشوائية مثلا .. إذا قام المستخدم بأخذ مجال ضيق للأعداد الصحيحة و قام بتوليد الكثير من الأرقام العشوائية ثم قام بتكرار العملية عدة مرات .. فسيلاحظ انتظام هذه الأعداد بشكل أو بآخر !!!
لذلك قام المبرمجون بإنشاء الدالة srand التي تعتمد على توليد الأرقام العشوائية حسب البذرة (seed) والبذرة عبارة عن عدد صحيح , هذه البذرة يجب أن تكون متغيرة في كل وقت .. لكي يكون العدد المولد "شبه عشوائي" , و عادة ما يستخدم المبرمجون "الزمن" كبذرة للدالة srand , يعني نُسند قيمة الثواني إلى الدالة srand فنحصل على رقم عشوائي و عندما نُسند قيمة الزمن مرة أخرى سنحصل على عدد عشوائي آخر ! لأن الوقت يتغير من لحظة إلى أخرى.
و لكن كيف يمكننا إسناد قيمة الثواني ؟
يتم ذلك باستخدام الدالة time الموجودة في المكتبة ctime , هذه الدالة تستقبل الوسيط المعدوم NULL أو يمكن أن نكتب العدد 0 ولكنني أفضل الكتابة الأولى.
الدالة time تستقبل الوسيط NULL و تعيد عدد الثواني التي مرت من الساعة: 00:00 في اليوم 1 جانفي في عام 1970.
طبعا يمكننا الآن أن نستنتج عدد الدقائق و الساعات والأيام و الشهور و ... إلخ
انظر الكود:

```
long seconds,minutes,days,hours;
seconds=time(NULL);
minutes=seconds/60;
hours=minutes/60;
days=hours/60;
```

و يمكننا أن نكتب كلمة time_t بدل الكلمة long لأن المكتبة ctime تحتوي على المتغير time_t المعرف كما يلي:

```
typedef long time_t;
```

الآن سنقوم بكتابة كود يخرج عددا عشوائيا:

```
#include<iostream>
#include<ctime>
#include<cstdlib>
using namespace std;
int main()
{
    srand(time(NULL));
    int n=rand();
    cout<<n<<endl;
    system("pause");
    return 0;
}
```

لاحظ أننا قمنا أولا بتجهيز الدالة srand عن طريق البذرة (seed) و من ثم قمنا بإسناد القيمة التي تعيدها الدالة rand إلى المتغير n.

حسنا .. الآن فهمنا كيفية توليد رقم عشوائي باستخدام الدالة rand و بالإعتماد على الدالة srand التي تحتاج إلى البذرة seed و التي عادة ما تكون "الزمن" أو لنقل:عدد الثواني المُعاد من طرف الدالة time.

الآن يمكننا استنتاج كيفية توليد أرقام عشوائية محصورة بين عدد معين يُدخله المستخدم و الصفر .. كيف ذلك؟

كل ما في الأمر أننا سنستخدم الرمز % و الذي يعني باقي القسمة و يخضع للقانون التالي:

باقي قسمة عدد a على b يكون دائما أصغر تماما من b و أكبر أو يساوي 0

لنسمي r باقي قسمة a على b و q ناتج القسمة , إذا: $a = bq + r$

حيث : r محصور بين 0 و b

و تسمى هذه النظرية بنظرية "اقليدس" وهو العالم الرياضي المشهور.

مثال:

```
#include<iostream>
#include<ctime>
#include<cstdlib>
using namespace std;
int main()
{
    int n;
    srand(time(NULL));
    cout<<"n=";
    cin>>n;
    cout<<rand()%n<<endl;
    system("pause");
    return 0;
}
```

و يمكننا توليد العديد من الأعداد العشوائية باستخدام الحلقة for.

طيب .. كيف يمكننا توليد أرقام عشوائية محصورة بين عددين يُدخلهما المستخدم ؟؟؟

قلنا سابق أنه يجب علينا إتباع قانون معقد بعض الشيء .. و القانون الذي سأستعمله هو:

ملاحظة:القانون مكتوب باللغة الفرنسية ولكني شرحته باللغة العربية.

```
Le nombre aléatoire =la valeur initial+rand()%(la valeur scalaire);
La valeur scalaire=la valeur final-la valeur initial+1;
```

حيث:

العدد العشوائي:le nombre aléatoire .

العدد الأول أو القيمة الإبتدائية:la valeur initial.

القيمة الموجهة:La valeur scalaire .

العدد الثاني أو القيمة النهائية:la valeur final .

طريقة توليد حروف عشوائية:

```
#include<iostream>
#include<ctime>
#include<string>
using namespace std;
int main()
{
    string str="ABCDEFGHIJKLMNOPQRST";
    srand(time(NULL));
    cout<<str[rand()%20]<<endl;
    system("pause");
    return 0;
}
```

في الحقيقة .. فإن هذا المثال بسيـــــــط و لا يحتاج إلى شرح و لكن سنمر عليه مر الكرام.

في البداية قمنا بالإعلان عن سلسلة حرفية باسم str ثم قمنا بتوليد رقم عشوائي يقع في النطاق المسموح به .. أي ما بين 0 و 19 و سبب ذلك هو أنه إذا كان العدد العشوائي يأخذ قيمة أكبر من 19 أو أصغر من الصفر فسننتقل إلى حدود خارج نطاق المصفوفة مما يسبب ظاهرة الفيض الحسابي او ما يعرف بـــ overflow لأن عناصر المصفوفة تمتلك الدليل من 0 و حتى 19 , لاحظ أن باقي القسمة على 20 يأخذ نفس المجال الذي يأخذه دليل عناصر المصفوفة.

الفكرة هنا تكمن في توليد رقم عشوائي يقع ضمن المجال الذي يتحرك فيه دليل عناصر المصفوفة ثم نقوم بطباعة العنصر الموافق للرقم العشوائي المُوَلّد .. مثلا إذا كان الرقم المولد هو 2 فسيظهر الحرف C لأنه يأخذ الترتيب 2 في المصفوفة.

4-كيف ننشئ دالة عشوائية ؟

رأينا سابقا كيفية توليد أرقام عشوائية باستخدام دوال جاهزة .. ولكن كيف يمكننا إنشاء دالة تقوم بتوليد الأعداد العشوائية ؟ و ما هي أهم الخوارزميات المستخدمة في هذا المجال ؟

في الحقيقة هناك بعض الأمور التي صنفها علماء الفيزياء على أنها عشوائية مثل تغير المقاومة واحتمال وجود "فوتون" في مكان معين ولكن دمجها في المعلوماتية في غاية الصعوبة .. إن لم يكن مستحيـــــــل !!! لذلك تم اللجوء إلى التفكير في الخوارزميات وأول خوارزمية طُرحت كانت من طرف العالم الشهير:

"Von Neuman" وهو أحد رواد المعلوماتية.

الخوارزمية التي اكتشفها "فون نيمان" تعتمد على اختيار عدد معين و رفعه للقوة 2 واختيار رقمين من الوسط في الناتج.

مثال:نختار الرقم 35 , بعد رفعه للقوة 2 سنجد العدد 1225 نختار رقمين من وسط العدد 1225 وهما 22 , نعاود نفس العملية .. نرفع 22 للقوة 2 فنحصل على العدد 484 نلاحظ أن العدد يتكون من 3 أرقام , نضيف 0 من اليسار ليصبح عدد أرقام العدد عبارة عن 4 كالتالي 0484 الآن نختار رقمين من الوسط و هما 48 , إذا استمرينا على نفس الحال فسنصل إلى هده المتتالية التي تشكل أرقاما عشوائية: 35 22 48 30 90 10

ولكن إذا جربنا الآن أن نقوم برفع 10 إلى القوة 2 واختيار رقمين فإننا دائما نعود إلى الرقم 10 وهذه مشكلة .. ولكن لا بأس بانطلاقة كهذه ...

و لكي تكون الخوارزمية أكثر تعقيدا !!! يقوم المبرمجون بالإعتماد على الوقت كبذرة لتوليد الأعداد العشوائية (كما قلنا سابقا).

بلا شك هده ليست أفضل خوارزمية ممكنة .. ولكنها مهمة لأنها أعطت الانطلاقة .. فظهرت عدة خوارزميات .. مثل الخوارزمية التالية:

```
#include<iostream>
#include<ctime>
using namespace std;
int main()
{
    unsigned int a=time(NULL),b=time(NULL),m=100;
    unsigned int nombre=time(NULL);
    for(int i=0;i<20;i++)
    {
        nombre=(a*nombre+ b)%m;
        cout<<nombre<<endl;
    }
    system("pause");
    return 0;
}
```

هذا البرنامج سوف يطبع 20 رقما عشوائيا وبالنسبة للخوارزمية فهي تعتمد على أخذ رقم معين يتم الحصول عليه من الزمن وبعد ذالك يتم ضربه في أحد الأرقام العشوائية التي اخترناها في البداية ثم إضافته إلى رقم عشوائي آخر .. و أخيرا نقوم بحساب باقي القسمة على الرقم 100 لأنني أردت إظهار أعداد عشوائية أقل من 100 و يمكنك تغيير العدد m كيفما تشاء .. كما يمكنك أن تجعله متغير عشوائي! كأن تسند إليه القيمة المعادة من طرف الدالة time مثلا و لكن ستظهر لكَ أعداد ضخمة.

تم الإنتهاء من الكتاب بتاريخ 2010/03/29